

**Application No. 09/256,845**

**Atty Docket: PUMA 1000-1**

**In the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (cancelled)
2. (previously presented) The method of claim 46, wherein the at least one template comprises at least one page description language template.
3. (previously presented) The method of claim 2, wherein said at least one page description language template comprises at least one Hypertext Markup Language (HTML) document.
4. (previously presented) The method of claim 2, wherein said at least one page description language template comprises at least one Standard Generalized Markup Language (SGML) document.
5. (previously presented) The method of claim 46, wherein the references are embedded in the at least one template using user-defined tags.
6. (previously presented) The method of claim 46, wherein the back-end information access functionality that is actually invoked is determined based, at least in part, on which platform a given client executes.
7. - 9. (cancelled)
10. (previously presented) The method of claim 46, wherein a template manager stores parsed versions of templates in a template cache, so that each template need only be parsed once.

**Applicati n No. 09/256,845**

**Atty Docket: PUMA 1000-1**

11. (previously presented) Th method of claim 10, wherein the parsed versions of templates are maintained on a persistent storage, so that the parsed templates are available from one session to another.

12. (previously presented) The method of claim 10, wherein at least one parsed version of a template is flushed, so that said system is forced to again parse the at least one template.

13. (cancelled)

14. (previously presented) The method of claim 47, wherein the back-end database includes an SQL database system that retrieves information in response to SQL queries.

15. - 18. (cancelled)

19. (previously presented) The method of claim 46, wherein the back-end information access functionality invoked is based, at least in part, on a specific client session that is executing.

20. (previously presented) The method of claim 46, wherein logic that implements the action of providing the specified functionality to access information consists of a single code base application that is deployed on multiple platforms.

21. (cancelled)

22. (previously presented) The method of claim 46, wherein the abstract references include tokens specifying programming constructs.

23. (original) The method of claim 22, wherein said programming constructs include conditional logic statements.

**Applicati n No. 09/256,845**

**Atty D cket: PUMA 1000-1**

24. (original) The method of claim 23, wh rein said conditional logic statements include "if" statements.

25. (original) The method of claim 23, wherein said conditional logic statements include "for" loops.

26. - 29. (cancelled)

30. (previously presented) The system of claim 60, wherein the particular template includes at least one page description language template.

31. (original) The system of claim 30, wherein said at least one page description language template comprises a Hypertext Markup Language (HTML) document.

32. (original) The system of claim 30, wherein said at least one page description language template comprises a Standard Generalized Markup Language (SGML) document.

33. (previously presented) The system of claim 60, wherein the abstract references are embedded in the particular template using user-defined tags.

34. (previously presented) The system of claim 60, wherein which run-time services that are actually invoked is determined based, at least in part, on which platform a given client executes.

35. (cancelled)

36. (previously presented) The system of claim 60, wherein said template manager stores parsed templates in a template cache, so that each template need only be parsed once.

**Applicati n No. 09/256,845**

**Atty D cket: PUMA 1000-1**

37. (previously presented) The system of claim 36, wherein said parsed templates are maintained on a persistent storage, so that the parsed templates are available from one application execution session to another.

38. (previously presented) The system of claim 36, wherein any parsed templates are occasionally flushed, so that said system is forced to again parse the particular template.

39. (cancelled)

40. (previously presented) The system of claim 60, wherein the back-end database comprises an SQL database system that retrieves information in response to SQL queries.

41. (previously presented) The system of claim 60, wherein the particular template comprises at least one read-only template.

42. (previously presented) The system of claim 60, wherein the particular template is loaded by browser software running at said particular client.

43. (previously presented) The system of claim 60, wherein the particular template comprises an input form having a platform-specific presentation when rendered at a given client.

44. (cancelled)

45. (previously presented) The system of claim 60, wherein which run-time services are invoked is determined based, at least in part, on a specific client session that is executing.

**Application No. 09/256,845**

**Atty Docket: PUMA 1000-1**

46. (previously presented) A method of creating and deploying an application that provides access to back-end information access functionality, including:

- creating at least one template including one or more abstract references that specify functionality to be invoked when a given client requests the template;
- registering the abstract references with a dictionary that associates the abstract references with at least one run-time handler and one or more run-time services; and

providing the specified functionality to access information, including:

- receiving a request from the given client that identifies at least one template;
- accessing the identified template and determining the abstract references in the identified template;
- accessing the dictionary and determining the run-time handler and the run-time services associated with the abstract references; and
- invoking the run-time handler and the run time services to access to the back-end information access functionality.

47. (previously presented) The method of claim 46, wherein the back-end functionality includes accessing information in a back-end database.

48. (previously presented) The method of claim 46, wherein the back-end functionality includes accessing information in a configuration table.

49. (previously presented) The method of claim 46, wherein the back-end functionality includes accessing information from machine services.

50. (previously presented) The method of claim 46, wherein the request can be resolved to the given client, further including invoking the run-time handler and the run time services using parameters corresponding to the client identity.

**Application No. 09/256,845**

**Atty Dock 1: PUMA 1000-1**

51. (previously presented) The method of claim 46, wherein the request can be resolved to a platform from which the request originates, further including invoking the run-time handler and the run time services using parameters corresponding to the platform.

52. (previously presented) The method of claim 46, wherein the request can be resolved to the given client and a platform from which the request originates, further including invoking the run-time handler and the run time services using parameters corresponding to the given client and the platform.

53. (previously presented) The method of claim 47, wherein the request can be resolved to the given client and a platform from which the request originates, further including invoking the run-time handler and the run time services using parameters corresponding to the given client and the platform.

54. (previously presented) The method of claim 46, wherein the abstract references specify functionality that is independent of a platform from which the request originates.

55. (previously presented) The method of claim 46, wherein the abstract references specify functionality that is independent of the given client.

56. (previously presented) The method of claim 46, wherein the abstract references specify functionality that is independent of any particular back-end database.

57. (previously presented) The method of claim 46, wherein the back-end database includes a synchronization engine.

58. (previously presented) The method of claim 46, wherein the request can be resolved to a platform from which the request originates, further including composing a presentation adapted to the platform.

**Application No. 09/256,845**

**Atty Docket: PUMA 1000-1**

59. (previously presented) The method of claim 58, wherein the run time services that access information in the back-end database are independent of logic that composes the adapted presentation.

60. (previously presented) A template repository and template manager system that provide access to a back-end information access functionality in response to a client request for a template, including:

- a template repository that stores templates, a particular template including one or more abstract references that specify back-end information access functionality to be invoked when the client requests the particular template;

- a dictionary that associates the abstract references with one or more run-time services; and

- a template manager, responsive to the client request that identifies the particular template, including logic that

- accesses the template repository and parses the particular template,
  - accesses the dictionary and resolves the abstract references to the associated run-time services, and

- invokes the associated run-time services that provide back-end information access functionality.